RFAssign1

Team IoT

14/02/2022

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.2

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.5 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.0.2 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## Warning: package 'tidymodels' was built under R version 4.1.2

## Registered S3 method overwritten by 'tune':  
## method from   
## required\_pkgs.model\_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --

## v broom 0.7.9 v rsample 0.1.1   
## v dials 0.0.10 v tune 0.1.6   
## v infer 1.0.0 v workflows 0.2.4   
## v modeldata 0.1.1 v workflowsets 0.1.0   
## v parsnip 0.1.7 v yardstick 0.0.9   
## v recipes 0.1.17

## Warning: package 'dials' was built under R version 4.1.2

## Warning: package 'infer' was built under R version 4.1.2

## Warning: package 'modeldata' was built under R version 4.1.2

## Warning: package 'parsnip' was built under R version 4.1.2

## Warning: package 'recipes' was built under R version 4.1.2

## Warning: package 'rsample' was built under R version 4.1.2

## Warning: package 'tune' was built under R version 4.1.2

## Warning: package 'workflows' was built under R version 4.1.2

## Warning: package 'workflowsets' was built under R version 4.1.2

## Warning: package 'yardstick' was built under R version 4.1.2

## -- Conflicts ----------------------------------------- tidymodels\_conflicts() --  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()  
## \* Use suppressPackageStartupMessages() to eliminate package startup messages

library(caret)

## Warning: package 'caret' was built under R version 4.1.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity

## The following object is masked from 'package:purrr':  
##   
## lift

library(gridExtra)

## Warning: package 'gridExtra' was built under R version 4.1.2

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(vip)

## Warning: package 'vip' was built under R version 4.1.2

##   
## Attaching package: 'vip'

## The following object is masked from 'package:utils':  
##   
## vi

library(ranger)

## Warning: package 'ranger' was built under R version 4.1.2

library(readr)

drug = read\_csv("drug\_data-1.csv")

## Rows: 1885 Columns: 32

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (19): Column14, Column15, Column16, Column17, Column18, Column19, Column...  
## dbl (13): Column1, Column2, Column3, Column4, Column5, Column6, Column7, Col...

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

names(drug) = c("ID", "Age", "Gender", "Education", "Country", "Ethnicity",  
"Nscore", "Escore", "Oscore", "Ascore", "Cscore", "Impulsive",  
"SS", "Alcohol", "Amphet", "Amyl", "Benzos", "Caff", "Cannabis",  
"Choc", "Coke", "Crack", "Ecstasy", "Heroin", "Ketamine", "Legalh",  
"LSD", "Meth", "Mushrooms", "Nicotine", "Semer", "VSA")  
str(drug)

## spec\_tbl\_df [1,885 x 32] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ID : num [1:1885] 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : num [1:1885] 0.4979 -0.0785 0.4979 -0.952 0.4979 ...  
## $ Gender : num [1:1885] 0.482 -0.482 -0.482 0.482 0.482 ...  
## $ Education: num [1:1885] -0.0592 1.9844 -0.0592 1.1637 1.9844 ...  
## $ Country : num [1:1885] 0.961 0.961 0.961 0.961 0.961 ...  
## $ Ethnicity: num [1:1885] 0.126 -0.317 -0.317 -0.317 -0.317 ...  
## $ Nscore : num [1:1885] 0.313 -0.678 -0.467 -0.149 0.735 ...  
## $ Escore : num [1:1885] -0.575 1.939 0.805 -0.806 -1.633 ...  
## $ Oscore : num [1:1885] -0.5833 1.4353 -0.8473 -0.0193 -0.4517 ...  
## $ Ascore : num [1:1885] -0.917 0.761 -1.621 0.59 -0.302 ...  
## $ Cscore : num [1:1885] -0.00665 -0.14277 -1.0145 0.58489 1.30612 ...  
## $ Impulsive: num [1:1885] -0.217 -0.711 -1.38 -1.38 -0.217 ...  
## $ SS : num [1:1885] -1.181 -0.216 0.401 -1.181 -0.216 ...  
## $ Alcohol : chr [1:1885] "CL5" "CL5" "CL6" "CL4" ...  
## $ Amphet : chr [1:1885] "CL2" "CL2" "CL0" "CL0" ...  
## $ Amyl : chr [1:1885] "CL0" "CL2" "CL0" "CL0" ...  
## $ Benzos : chr [1:1885] "CL2" "CL0" "CL0" "CL3" ...  
## $ Caff : chr [1:1885] "CL6" "CL6" "CL6" "CL5" ...  
## $ Cannabis : chr [1:1885] "CL0" "CL4" "CL3" "CL2" ...  
## $ Choc : chr [1:1885] "CL5" "CL6" "CL4" "CL4" ...  
## $ Coke : chr [1:1885] "CL0" "CL3" "CL0" "CL2" ...  
## $ Crack : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ Ecstasy : chr [1:1885] "CL0" "CL4" "CL0" "CL0" ...  
## $ Heroin : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ Ketamine : chr [1:1885] "CL0" "CL2" "CL0" "CL2" ...  
## $ Legalh : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ LSD : chr [1:1885] "CL0" "CL2" "CL0" "CL0" ...  
## $ Meth : chr [1:1885] "CL0" "CL3" "CL0" "CL0" ...  
## $ Mushrooms: chr [1:1885] "CL0" "CL0" "CL1" "CL0" ...  
## $ Nicotine : chr [1:1885] "CL2" "CL4" "CL0" "CL2" ...  
## $ Semer : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ VSA : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Column1 = col\_double(),  
## .. Column2 = col\_double(),  
## .. Column3 = col\_double(),  
## .. Column4 = col\_double(),  
## .. Column5 = col\_double(),  
## .. Column6 = col\_double(),  
## .. Column7 = col\_double(),  
## .. Column8 = col\_double(),  
## .. Column9 = col\_double(),  
## .. Column10 = col\_double(),  
## .. Column11 = col\_double(),  
## .. Column12 = col\_double(),  
## .. Column13 = col\_double(),  
## .. Column14 = col\_character(),  
## .. Column15 = col\_character(),  
## .. Column16 = col\_character(),  
## .. Column17 = col\_character(),  
## .. Column18 = col\_character(),  
## .. Column19 = col\_character(),  
## .. Column20 = col\_character(),  
## .. Column21 = col\_character(),  
## .. Column22 = col\_character(),  
## .. Column23 = col\_character(),  
## .. Column24 = col\_character(),  
## .. Column25 = col\_character(),  
## .. Column26 = col\_character(),  
## .. Column27 = col\_character(),  
## .. Column28 = col\_character(),  
## .. Column29 = col\_character(),  
## .. Column30 = col\_character(),  
## .. Column31 = col\_character(),  
## .. Column32 = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

drug[drug == "CL0"] = "No"  
drug[drug == "CL1"] = "No"  
drug[drug == "CL2"] = "Yes"  
drug[drug == "CL3"] = "Yes"  
drug[drug == "CL4"] = "Yes"  
drug[drug == "CL5"] = "Yes"  
drug[drug == "CL6"] = "Yes"  
  
drug\_clean = drug %>% mutate\_at(vars(Age:Ethnicity), funs(as\_factor)) %>%  
mutate(Age = factor(Age, labels = c("18\_24", "25\_34", "35\_44",  
"45\_54", "55\_64", "65\_"))) %>%  
mutate(Gender = factor(Gender, labels = c("Male", "Female"))) %>%  
mutate(Education = factor(Education, labels =  
c("Under16", "At16", "At17", "At18", "SomeCollege",  
"ProfessionalCert", "Bachelors", "Masters", "Doctorate"))) %>%  
mutate(Country = factor(Country,  
labels = c("USA", "NewZealand", "Other", "Australia",  
"Ireland","Canada","UK"))) %>%  
mutate(Ethnicity = factor(Ethnicity,  
labels = c("Black", "Asian", "White", "White/Black", "Other",  
"White/Asian", "Black/Asian"))) %>%  
 mutate\_at(vars(Alcohol:VSA), funs(as\_factor)) %>%  
select(-ID)

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

str(drug\_clean)

## tibble [1,885 x 31] (S3: tbl\_df/tbl/data.frame)  
## $ Age : Factor w/ 6 levels "18\_24","25\_34",..: 3 2 3 1 3 6 4 3 3 5 ...  
## $ Gender : Factor w/ 2 levels "Male","Female": 2 1 1 2 2 2 1 1 2 1 ...  
## $ Education: Factor w/ 9 levels "Under16","At16",..: 6 9 6 8 9 4 8 2 6 8 ...  
## $ Country : Factor w/ 7 levels "USA","NewZealand",..: 7 7 7 7 7 6 1 7 6 7 ...  
## $ Ethnicity: Factor w/ 7 levels "Black","Asian",..: 6 3 3 3 3 3 3 3 3 3 ...  
## $ Nscore : num [1:1885] 0.313 -0.678 -0.467 -0.149 0.735 ...  
## $ Escore : num [1:1885] -0.575 1.939 0.805 -0.806 -1.633 ...  
## $ Oscore : num [1:1885] -0.5833 1.4353 -0.8473 -0.0193 -0.4517 ...  
## $ Ascore : num [1:1885] -0.917 0.761 -1.621 0.59 -0.302 ...  
## $ Cscore : num [1:1885] -0.00665 -0.14277 -1.0145 0.58489 1.30612 ...  
## $ Impulsive: num [1:1885] -0.217 -0.711 -1.38 -1.38 -0.217 ...  
## $ SS : num [1:1885] -1.181 -0.216 0.401 -1.181 -0.216 ...  
## $ Alcohol : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Amphet : Factor w/ 2 levels "Yes","No": 1 1 2 2 2 2 2 2 2 2 ...  
## $ Amyl : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Benzos : Factor w/ 2 levels "Yes","No": 1 2 2 1 2 2 2 2 2 2 ...  
## $ Caff : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Cannabis : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 1 1 1 1 1 ...  
## $ Choc : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Coke : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 1 1 1 1 ...  
## $ Crack : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ecstasy : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Heroin : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ketamine : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 1 1 1 1 ...  
## $ Legalh : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LSD : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Meth : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Mushrooms: Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 1 1 1 ...  
## $ Nicotine : Factor w/ 2 levels "Yes","No": 1 1 2 1 1 1 1 2 1 1 ...  
## $ Semer : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ VSA : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...

## Including Plots

You can also embed plots, for example:

drug\_clean = drug\_clean %>% select(!(Alcohol:Mushrooms)) %>% select(!(Semer:VSA))  
names(drug\_clean)

## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

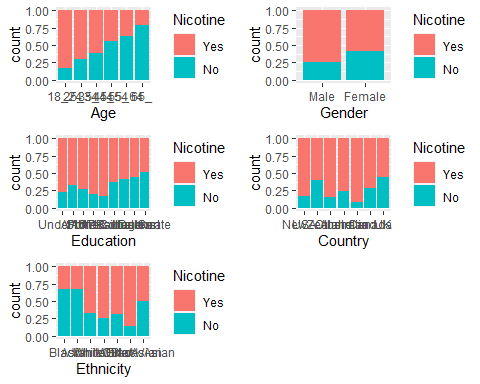
Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

# No missing data

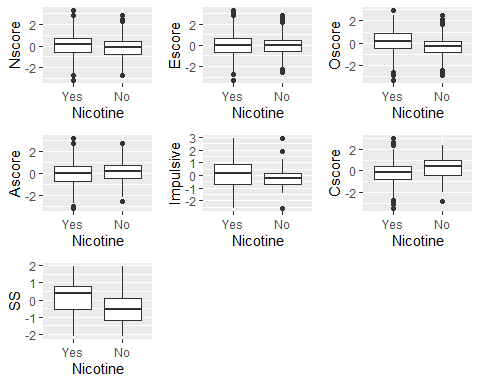
Now we’ll split the data.

set.seed(123)   
churn\_split = initial\_split(drug\_clean, prop = 0.7, strata = Nicotine) #70% in training  
train = training(churn\_split)  
test = testing(churn\_split)

p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill")  
p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
  
grid.arrange(p1,p2,p3,p4,p5)



p1 = ggplot(train, aes(x = Nicotine, y = Nscore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y = Escore)) + geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y = Oscore)) + geom\_boxplot()  
p4 = ggplot(train, aes(x = Nicotine, y = Ascore)) + geom\_boxplot()  
p5 = ggplot(train, aes(x = Nicotine, y = Impulsive)) + geom\_boxplot()  
p6 = ggplot(train, aes(x = Nicotine, y = Cscore)) + geom\_boxplot()  
p7 = ggplot(train, aes(x = Nicotine, y = SS)) + geom\_boxplot()  
grid.arrange(p1,p2,p3,p4, p5, p6, p7, ncol = 3)

 Set up our folds for cross-validation

set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

drug\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>% #add tuning of mtry and min\_n parameters  
 #setting trees to 100 here should also speed things up a bit, but more trees might be better  
 set\_engine("ranger", importance = "permutation") %>% #added importance metric  
 set\_mode("classification")  
  
drug\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_recipe)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(2, 8)), #these values determined through significant trial and error  
 min\_n(range = c(5, 20)), #these values determined through significant trial and error  
 levels = 10  
)  
  
set.seed(123)  
rf\_res\_tuned = tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = rf\_grid #use the tuning grid  
)